



## **LibreSource: Une plate-forme de développement collaboratif basée sur le WEB**

Florent Jouille, Bernard Ganne, Hala Skaf-Molli, Pascal Molli

### **► To cite this version:**

Florent Jouille, Bernard Ganne, Hala Skaf-Molli, Pascal Molli. LibreSource: Une plate-forme de développement collaboratif basée sur le WEB. 19èmes Journées Internationales "Génie Logiciel & Ingénierie de Systèmes et leurs Applications" - ICSSEA 2006, Dec 2006, Paris/France. inria-00105542

**HAL Id: inria-00105542**

**<https://hal.inria.fr/inria-00105542>**

Submitted on 11 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## LibreSource

Une plate-forme de développement collaboratif basée sur le WEB  
Un produit de la recherche publique industrialisé par la société Artenum

*Florent Jouille, Bernard Ganne, Hala Skaf-Molli, Pascal Molli*

[jouille, ganne, skaf, molli]@loria.fr



### **Résumé**

L'objectif principal de cet article est de décrire une technologie de collaboration appelée LibreSource<sup>1</sup>. LibreSource a été conçu pour pallier aux problèmes de centralisation et de verrouillage des données dans les systèmes coopératifs existants. C'est une plate-forme ouverte, elle fournit tous les services nécessaires pour le travail coopératif tel que le partage de données, la coordination, et la communication. Cet article présente l'approche de LibreSource, son architecture et montre son utilisation dans le cadre d'un procédé coopératif de développement logiciel.

*The existing collaborative distributed environments have several important drawbacks. Consequently, users need to rely on very few of installation like SourceForge and accept all of the associated risks. The aim of this article is to describe a collaborative platform called LibreSource. LibreSource as been developed in order to support large scale collaborative work. This platform is an open source software that provides framework and the associated services designed to support different kind of collaborative processes.*

---

<sup>1</sup> LibreSource, disponible sur <http://www.libresource.org>

## Introduction

Ces dernières années, beaucoup de plates-formes de collaboration basées sur INTERNET ont émergé. Elles peuvent être classifiées en deux grandes catégories :

- Plates-formes orientées *gestion de communautés* qui sont souvent représentées par des systèmes de gestion de contenus comme Zope/Plone<sup>2</sup> et PhpNuke<sup>3</sup>. L'objectif de ces plates-formes est d'assurer le flux d'information et la communication entre plusieurs groupes ou plusieurs personnes.
- Plates-formes orientées *production* qui sont représentées par des environnements de développement de logiciels. SourceForge<sup>4</sup> et ses descendants Gforge<sup>5</sup>, Savannah<sup>6</sup> sont de bons représentants de ce genre de plates-formes (voir aussi [1]).

Tous ces environnements fournissent plus ou moins le même genre de fonctionnalités ou de services. Les gestionnaires de contenu proposent des outils de communication (listes de diffusion, forums de discussion) et de publication de contenu (pages wiki, partage de fichiers), tandis que les plates-formes orientées production permettent le partage de dossiers et de fichiers (gestion de configuration, gestion de zones de travail), ainsi que la communication et la coordination (gestion de tâches). Mais malgré leur large diffusion et leur popularité, ces outils sont limités par deux principaux inconvénients : le verrouillage des données et la centralisation des services.

Cet article introduit LibreSource, une plate-forme distribuée dédiée au travail collaboratif. LibreSource a été spécialement développé pour supporter un grand nombre de structures de collaboration, en étant adaptable à différents types de communautés et accessible à des utilisateurs non spécialistes. De plus, comparé aux environnements existants et de même catégorie (GForge et Savannah), LibreSource propose un haut niveau d'intégration qui permet de palier aux inconvénients cités ci-dessus.

Cet article est construit comme suit. Dans un premier temps, nous montrerons les avantages de l'architecture de LibreSource. Puis nous détaillerons cette architecture et les services actuellement proposés par la plate-forme. Enfin, nous montrerons un exemple d'utilisation concret à travers un projet de développement logiciel.

## 1. L'approche de LibreSource

La *collaboration* est ici décrite comme l'activité d'un groupe contenant un grand nombre de participants (par exemple une communauté), qui doivent atteindre un certain but. Une *structure de collaboration* est vue comme un outil orienté internet qui supporte la collaboration. En outre, une structure de collaboration *complexe* est une agrégation de différents outils qui permettent collaboration et communication. De plus, nous partons du principe que la communauté est distribuée géographiquement. Un exemple bien connu est la communauté SourceForge, qui regroupe des développeurs du monde entier sur des projets de logiciels libres.

Comme annoncé précédemment, les environnements de collaboration existants ont deux inconvénients majeurs :

**Données verrouillées** - C'est un problème important. En effet, il est difficile de créer un projet sous un serveur SourceForge par exemple et de le déplacer vers un autre serveur. Les utilisateurs sont en quelque sorte liés à un site SourceForge. Ce type de problème a été signalé depuis longtemps et des initiatives telles que CoopX et d'autres ont essayé de le résoudre.

**Centralisation** - Les environnements existants sont basés sur une architecture centralisée. Selon cette architecture, les données partagées ainsi que leur contrôle sont gérés par une seule machine : *le serveur*. Un des problèmes majeurs de cette architecture est la disponibilité des données. Si le serveur tombe en panne, alors les données ne sont plus accessibles. Par exemple, si SourceForge ferme, la communauté perd l'accès à 80 000 projets. Un autre problème important est le passage à l'échelle du serveur par rapport au nombre d'utilisateurs. Si le nombre d'utilisateurs augmente, alors le serveur peut devenir un point d'étranglement du réseau. Une solution possible est l'achat

---

<sup>2</sup> Zope/Plone, disponible sur <http://www.zope.org/>

<sup>3</sup> PhpNuke, disponible sur <http://phpnuke.org/>

<sup>4</sup> SourceForge, disponible sur <http://www.sf.net/>

<sup>5</sup> GForge, disponible sur <http://gforge.org/>

<sup>6</sup> Savannah, disponible sur <http://savannah.gnu.org/>

d'un nouveau serveur plus puissant, ceci signifie plus de dépenses. En plus du coût élevé du serveur, les coûts de son administration et de sa maintenance et les coûts de communication ne peuvent pas être partagés entre les différents partenaires.

Un dernier problème, tout aussi important, est la notion de possession - *ownership* - du serveur. Si le serveur est installé dans une entreprise, les autres partenaires peuvent subir la 'censure'. L'*ownership* peut limiter l'accès aux données, et ainsi empêcher certains partenaires de travailler correctement. Une solution possible à ce problème est de faire héberger le serveur chez un tiers de confiance. Ceci introduit des surcoûts et ne permet pas d'éviter les problèmes liés aux architectures centralisées. Par exemple, SourceForge a changé de licence, il est passé d'open source à une licence fermée. Ceci a mis en danger 80 000 projets.

Afin d'éviter les problèmes liés aux verrouillages de données et aux architectures centralisées existantes, LibreSource est basé sur la notion de *fédération*. En effet, LibreSource est conçu pour être déployé comme un réseau de serveurs. Ainsi, au lieu de construire un système complexe avec 100 000 utilisateurs, nous proposons d'agréger cent nœuds de milliers d'utilisateurs. L'infrastructure requise est beaucoup plus simple et la visibilité des nœuds est maintenue.

Cette approche permet d'éviter les deux principaux inconvénients cités ci-dessus : il est facile de déplacer un projet LibreSource d'un serveur vers un autre serveur. Elle améliore considérablement la performance en distribuant la charge globale sur les différents nœuds. Elle réduit aussi les problèmes de sécurité : si la sécurité d'un serveur est compromise, alors les autres serveurs peuvent continuer à fournir le service. Elle permet de partager les frais de service à travers une fédération et de réduire le risque de censure. Aujourd'hui, LibreSource est la seule plate-forme qui offre ces fonctionnalités.

## 2. L'architecture de LibreSource

Pour faciliter la compréhension de l'architecture de LibreSource, il est possible de comparer le comportement d'un serveur LibreSource à un système de fichiers ordinaire. Ainsi, un serveur LibreSource est un arbre de composants. Un composant peut-être un forum, un projet, un traqueur de bogues ou d'autres objets... Comme dans un système de fichiers, chaque composant de l'arbre LibreSource a sa propre politique de sécurité. Les utilisateurs peuvent créer un composant, le détruire, l'éditer ou le déplacer. En outre, LibreSource a une architecture ouverte. Les membres de la communauté peuvent développer leurs propres composants et les *intégrer* facilement dans l'arbre LibreSource.

Le système repose donc sur un noyau qui se charge de gérer l'arborescence des composants. On trouve à ce niveau toutes les fonctionnalités qui permettent de gérer la sécurité (donner ou retirer des droits d'accès, créer des groupes, rattacher des utilisateurs à des projets, etc.), l'indexation des données (utile au moteur de recherche intégré) et les événements (contexte d'exécution d'une action sur l'arbre des ressources). Ces derniers jouent un rôle important dans la création d'une conscience de groupe au sein de la communauté des utilisateurs. Celle-ci permet aux personnes impliquées dans un projet d'être informées des modifications de ce dernier, ce qui s'avère très utile lorsque l'on travaille en étant distribué géographiquement et/ou dans le temps.

### Organisation des composants

Les autres services viennent se greffer sur ce noyau de base pour proposer leurs services. LibreSource fournit plusieurs services dédiés à la coopération tels que le partage de données, la communication et la coordination. La conception de LibreSource permet de simplifier l'intégration de ces différents composants. L'architecture étant comparable à un système de fichiers ordinaire, on peut laisser le noyau se charger de gérer la sécurité, les événements et l'indexation au niveau des nœuds. La création d'un nouveau composant revient ainsi à créer une nouvelle instance du type correspondant et à le placer dans l'arbre LibreSource.

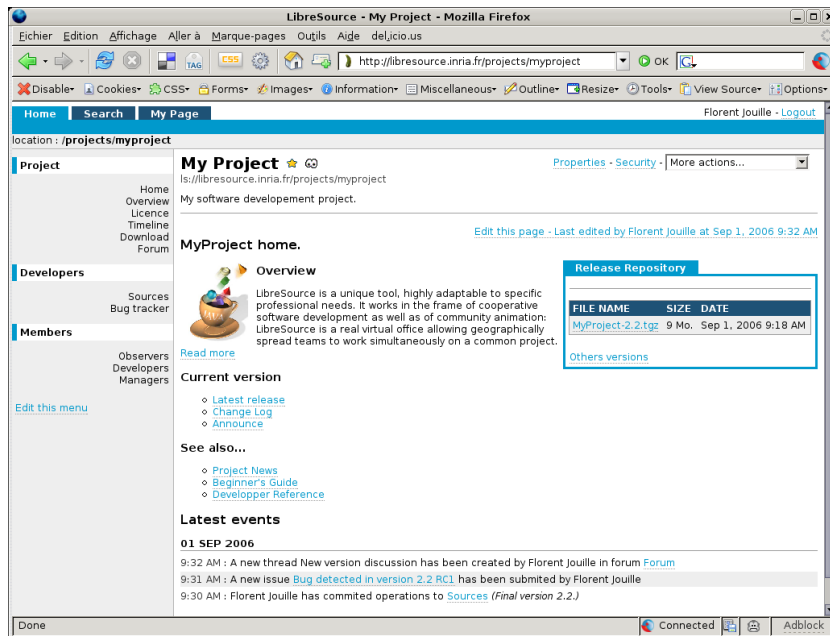


Figure 1 : Page d'accueil d'un projet sur LibreSource.

Nous décrivons ici les différents types de composants disponibles :

### Partage de données

Le partage des données est crucial pour le travail coopératif. Une des forces principales de LibreSource est que les données peuvent être propagées d'un serveur à un autre. Pour cela on emploie un composant de nouvelle génération – un synchroniseur – appelé So6. Plus précisément, So6 est un synchroniseur générique de fichiers, décrit dans [2].

#### Synchroniseur

Le synchroniseur est le composant qui permet le partage synchronisé de données. C'est la partie serveur de l'outil So6, spécialement développé pour la plate-forme (figure 2).

So6 est un gestionnaire de configuration basé sur les transformées opérationnelles. Son utilisation est détaillée dans la troisième partie de cet article.

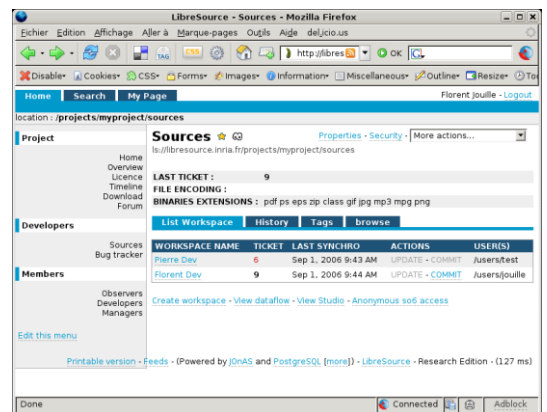


Figure 2 : La vue d'un synchroniseur.

### Espace de téléchargement

Pour publier des fichiers de manière non synchronisée, on peut instancier un espace de téléchargement. Un utilisateur peut y « uploader » un fichier, le rendant ainsi accessible aux autres utilisateurs de la plate-forme (figure 3).

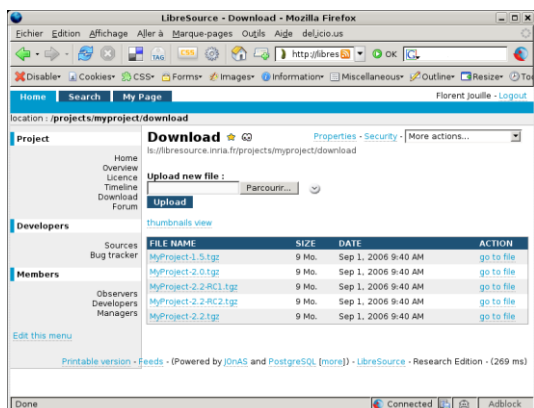


Figure 3 : Un espace de téléchargement.

### Pages wiki

Une page wiki permet de très simplement publier une page web. On utilise pour cela une syntaxe particulière appelée *wiki* qui permet de mettre en forme des pages XHTML sans connaître forcément ce langage. Des balises simplifiées, ainsi que des macros, permettent une mise en forme aisée. On peut ainsi insérer des images, des liens, formater le texte, etc. très facilement (*Figure 1*).

## Communication

### Forum

LibreSource intègre un gestionnaire de forum. C'est un composant à trois niveaux : un forum est composé d'une liste de sujets, qui sont eux-mêmes composés d'une suite de messages. Les messages supportent la syntaxe wiki.

### Messagerie électronique / Messagerie instantanée

Les utilisateurs peuvent fournir des informations personnelles, comme leur adresse de courrier électronique, ou leur identifiant de messagerie instantanée. Ces informations, outre le fait qu'elles soient visibles par les autres utilisateurs, sont utilisées par le serveur. Celui-ci peut automatiquement notifier les personnes de l'activité des projets auxquels elles prennent part.

### Liste de diffusion

Les listes de diffusion permettent de communiquer entre personnes d'un même projet.

## Coordination (aide à la décision)

### Traqueurs de bogues

Dans un processus de développement logiciel, le suivi des bogues est essentiel. LibreSource intègre un traqueur de bogues, qui permet l'ajout de nouvelles tâches (*figure 4*).

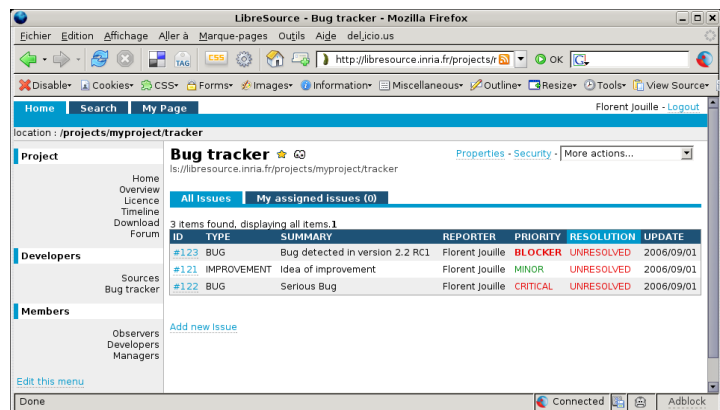


Figure 4 : Un traqueur de bogues.

L'avantage de LibreSource est que pour un même projet, plusieurs traqueurs peuvent être instanciés. On pourra ainsi en avoir un dédié aux améliorations, un autre aux bogues et un dernier aux tâches assignées aux membres du projet. La flexibilité de LibreSource permet de déplacer facilement une tâche d'un traqueur à l'autre.

### Module de vote

Pour permettre la coordination des membres et coordonner les choix à suivre lors de l'avancement d'un projet, on peut créer un sondage pour récolter les avis sur une question particulière. Cette fonctionnalité permet une aide à la décision.

### Gestion de procédés

L'intégration d'un gestionnaire de procédés dans LibreSource n'est pas encore réalisée. L'équipe ECOO, co-auteur de LibreSource s'intéresse aussi à ces problèmes de « workflow ». Un projet nommé Bonita<sup>7</sup> a ainsi été développé au sein de l'équipe, et il est prévu d'intégrer prochainement ses fonctionnalités dans la plate-

<sup>7</sup> Bonita, désormais supporté par Bull, est disponible sous licence LGPL à l'adresse <http://bonita.objectweb.org/>

forme LibreSource pour fournir une réelle gestion de procédés (publication de documents, cycle de développement logiciel, etc.).

## Conscience de groupe

Contrairement à un système de fichiers traditionnel, chaque noeud de l'arbre de ressources peut générer des événements en fonction de son activité. Ces événements sont très utiles pour tenir informé les membres de la plate-forme et ainsi créer une bonne conscience de groupe.

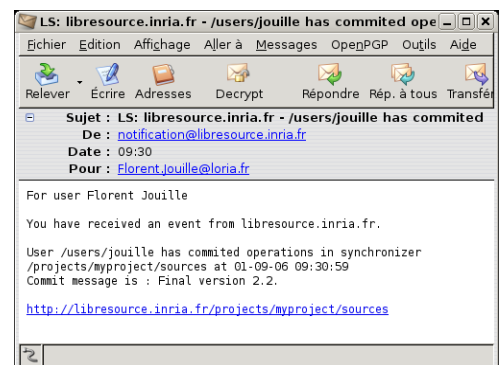
### Enregistreur d'événements

Un composant de LibreSource est dédié au stockage de ces événements. Facilement paramétrable, il permet d'avoir, sur la page d'accueil du projet par exemple, la liste des dernières actions réalisées dans le projet. On peut ainsi savoir qui a posté le dernier message dans un forum ou quand a été résolu un bogue.

### Notification

L'utilisateur peut aussi « s'abonner » aux événements qui l'intéressent en fonction de leur provenance (composant(s) concerné(s)) et/ou de leur type (action effectuée). Il peut également choisir le type de notification qui lui convient le mieux (courrier électronique ou messagerie instantanée). Il peut recevoir ainsi le détail de l'événement, avec des informations complémentaires (date et heure, utilisateur qui a effectué l'action, etc.).

Figure 5 : Exemple de notification par mail.



### Syndication

Un autre moyen donné aux utilisateurs pour rester en contact avec l'activité d'un projet est la syndication de certains composants. Elle se fait grâce à des flux de type RSS (*Really Simple Syndication* souscription vraiment simple), et permet, en utilisant un lecteur adapté, de connaître l'évolution de l'état de ces composants (ajout d'un patch dans un synchroniseur, nouveau fichier dans un espace de téléchargement, etc.).

## L'architecture distribuée et ses avantages

### Architecture à composants

Les plates-formes existantes sont des compositions de logiciels, et non de composants. L'avantage de LibreSource réside dans l'intégration des services sous forme de composants. Chaque composant fournit ses vues, ses événements, sa manière d'être importé/exporté, sa politique de sécurité, etc. Et c'est le *conteneur* qui gère ces fonctionnalités.

Grâce à cette architecture, on s'affranchit du problème de verrouillage des données. La possibilité de déplacer un projet et l'ensemble de ses données d'un serveur à l'autre permet aussi de fournir une meilleure disponibilité de service. De plus, l'architecture de LibreSource reste ouverte, le développement de nouveaux services étant possible et facilité.

### Architecture distribuée

La plupart des outils existants sont basés sur le modèle client/serveur, avec un site central hébergeant toutes les données. Ce modèle possède de gros inconvénients : tolérance aux pannes et aux attaques, performances, censure des données, etc.

LibreSource a été pensé avec une autre logique, celle de « réseaux de serveurs ». Pour l'instant, seules les données gérées par So6 peuvent être répliquées sur plusieurs sites LibreSource (cette fonctionnalité est détaillée dans la troisième partie de ce document). Mais on peut d'ores et déjà créer des réseaux de serveurs

LibreSource grâce à un composant particulier, qui permet le référencement sur le site local des projets hébergés sur d'autres sites. On a ainsi accès à la liste complète des projets d'une fédération de serveurs, et on peut y rechercher des mots clés de manière transparente. L'authentification centralisée est elle aussi supportée, grâce à l'utilisation d'un client CAS [3]. Une organisation peut installer plusieurs sites, tout en gardant sa gestion des comptes utilisateurs (sur un annuaire LDAP par exemple). Les utilisateurs peuvent ainsi se connecter sur tous les sites de la fédération avec les mêmes informations de session.

### 3. Un exemple

#### *Un processus de développement logiciel*

Nous illustrons dans cette partie l'utilisation de LibreSource au travers d'un projet de développement logiciel.

#### *Sécurité*

Le projet comporte trois groupes : les observateurs, les développeurs et les administrateurs. Les observateurs ont un accès en lecture seule sur le projet et ses ressources. Les développeurs possèdent en plus le droit de se connecter aux synchroniseurs et de commiter. Enfin, les administrateurs peuvent créer de nouvelles ressources et sont en charge de la mise à jour des ressources existantes.

#### *Partage synchronisé des données*

Le modèle mis en place est basé sur trois synchroniseurs. Le premier permet aux développeurs de partager les sources du logiciel. Le second est un synchroniseur de tests. Enfin, le dernier sert à publier les versions stables.

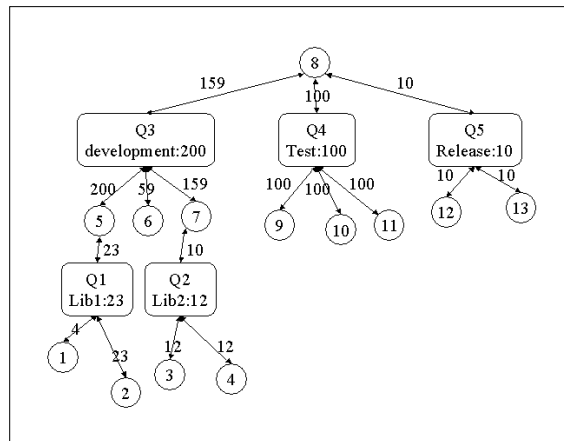


Figure 6 : Développement logiciel avec LibreSource.

La figure possède deux types de composants LibreSource : les Queues So6 (représentées par les rectangles) et les Espaces de travail (symbolisés par les ronds). Chaque queue contient  $n$  opérations (par exemple la queue  $Q3$  contient 200 opérations). Les espaces de travail génèrent des modifications (appelées opérations) et les transmettent aux Queues (action de « commiter »). Ils peuvent aussi se mettre à jour depuis les queues associées. Le nombre d'opérations échangées lors des propagation/mise à jour entre un espace de travail et la queue correspondante est inscrit sur la flèche les reliant.

On peut remarquer qu'un espace de travail peut être relié à plusieurs queues. Par exemple, l'espace de travail 8 est relié aux queues  $Q3$ ,  $Q4$  et  $Q5$ . Il a déjà reçu 159 opérations de la queue  $Q3$ , et est à jour avec les queues  $Q4$  et  $Q5$ . Les espaces de travail 5, 6 et 7 sont aussi connectés à la queue  $Q3$ . Dans le même temps, l'espace de travail 5 est à jour avec la queue  $Q1$ , alors que le numéro 7 doit encore intégrer des opérations provenant de la queue  $Q2$ .

Un espace de travail ne peut commiter de nouvelles opérations que s'il est à jour avec la queue correspondante. Cette solution permet de ne pas avoir de problèmes de mises à jour incomplètes ni de perte d'opérations. Les opérations reçues d'une queue sont intégrées et fusionnées aux éventuelles opérations locales grâce à l'algorithme des transformées opérationnelles. So6 assure ainsi la convergence des données



partagées (si tous les espaces de travail sont à jour et ont reçu toutes les modifications, ils contiennent les mêmes données). Donc, en utilisant So6, les utilisateurs propriétaires des espaces de travail envoient le flux des modifications qui est ensuite propagé aux autres queues.

Dans notre exemple de processus de développement logiciel décrit à la *figure 6*, on suppose que l'utilisateur à qui appartient l'espace de travail 8 est responsable de la propagation des modifications entre les développeurs, les testeurs, et la publication de la version stable. L'espace de travail 8 est ainsi mis à jour avec les modifications de la Queue 3 (développement). Ces modifications peuvent ainsi être propagées à la queue de test (Queue 4). Si les testeurs sont d'accord avec les modifications proposées, les modifications sont propagées à la Queue 5 (version stable) pour la sortie d'une nouvelle version du logiciel. Par contre, si les testeurs détectent des problèmes dans le nouveau code (bugs, violation de non-régression, etc.), un rapport peut être émis et envoyé aux développeurs. Cette opération peut elle-même être une modification du code (marquage des fragments qui posent problème), qui sera propagée aux développeurs par l'intermédiaire de l'espace de travail 8. Les développeurs pourront alors corriger le problème et renouveler la propagation de la nouvelle version.

Ce scénario nécessite trois serveurs LibreSource. Les Queues 1, 2 et 3 sont stockées sur un même serveur, et les Queues 4 et 5 sur des serveurs différents. Ainsi, les développeurs peuvent disposer de leur propre serveur, les testeurs également, tandis que la publication des versions stables se fera sur un serveur dédié, pouvant supporter une forte charge. D'autres configurations sont bien sûr possibles.

## Conclusion et perspectives

Cet article décrit la plate-forme distribuée LibreSource, qui a été développée pour supporter le travail coopératif dans de complexes structures de collaboration. Nous avons montré comment il est possible d'utiliser LibreSource dans le cadre d'un développement logiciel pour lequel la coordination des équipes de développement se fait de manière distribuée géographiquement et/ou dans le temps.

LibreSource est entièrement accessible avec un simple navigateur et peut être adapté très facilement à un grand nombre de communautés. On peut imaginer des utilisations variées, comme l'elearning, le développement logiciel ou simplement la publication en ligne.

Les perspectives sont un modèle entièrement distribué, déployé sur un réseau pair-à-pair. Ce cas de figure permettrait d'améliorer encore la tolérance aux pannes et la disponibilité des services hébergés, tout en supprimant complètement le risque de censure, les données étant répliquées sur chaque noeud du réseau.

## Références

- [1] CALIBRE (**Coordination Action for Libre Software Engineering for Open Development Platforms for Software and Services**), *Distributed Collaborative Development Environments*, 31/08/2005.
- [2] Molli, P., Oster, G., Skaf-Molli, H. and Imine, A. (2003), “**Using the Transformational Approach to Build a Safe and Generic Data Synchronizer**”, *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: Group2003*, Sanibel Island, Florida, USA.
- [3] Mathieu, V., Aubry, P. and Marchal, J. (2003), “**Single Sign-On open-source with CAS (Central Authentication Service)**”, *JRES2003*, Lille, France.

LibreSource a été financé de 2002 à 2004 par le RNTL (*Réseau National des Technologies Logicielles*). Il est co-développé par le LORIA (INRIA-Lorraine) et l'entreprise Artenum. Il est disponible sous deux formes : LibreSource Community Edition sous licence open source (QPL) et LibreSource Enterprise Edition. LibreSource a reçu le prix de ObjectWeb Jury's Special Price à *ObjectWebCon' 06*.